09/856936 PCT/DE99/03824

BUNDESREPUBLIK DEUTSCHLAND

DE 99/3824 EJU

0816 - 161

Bescheinigung

Die Siemens Aktiengesellschaft in München/Deutschland hat eine Patentanmeldung unter der Bezeichnung

"Verfahren und Anordnung zur Decodierung eines vorgegebenen Codeworts"

am 1. Dezember 1998 beim Deutschen Patent- und Markenamt eingereicht.

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

Die Anmeldung hat im Deutschen Patent- und Markenamt vorläufig die Symbole H 03 M und H 04 J der Internationalen Patentklassifikation erhalten.

München, den 21. März 2000

Deutsches Patent- und Markenamt

Der Präsident

Im Auftrag

Aktenzeichen: 198 55 453.2

Dzierzon

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH
RULE 17.1(a) OR (b)

Beschreibung

Verfahren und Anordnung zur Decodierung eines vorgegebenen Codeworts

Die Erfindung betrifft ein Verfahren und eine Anordnung zur Decodierung eines vorgegebenen Codeworts.

Bei der Decodierung eines Codeworts, das eine vorgegebene Anzahl Stellen aufweist, sollen die informationstragenden Stellen möglichst vollständig wiederhergestellt werden.

Die Decodierung findet auf der Seite des Empfängers statt, der das Codewort über einen gestörten Kanal empfangen hat. Signale werden insbesondere als Boolesche Werte, bevorzugt unterteilt in $+1$ und -1, über den Kanal übertragen, erfahren dort eine Störung und werden von einem Demodulator in analoge Werte umgesetzt, die mehr oder weniger stark von den vorgegebenen Booleschen Werten (± 1) abweichen können.

Allgemein wird ausgegangen von K Stellen binärer Information („Informationsbits") ohne Redundanz $u \in \{\pm 1\}^K$, die von einem Kanalcodierer mittels systematischen Blockcodes oder unsystematischen Blockcodes in ein Codewort $c \in \{\pm 1\}^N$ abgebildet wird. Dabei enthält das Codewort $N - K$ Bits (auch: „Prüfbits"), die als redundante Information zu den N Informationsbits zur Wiederherstellung der Information nach Übertragung über den gestörten Kanal einsetzbar sind.

Der systematische Blockcode fügt zu den N Informationsbits $N - K$ Prüfbits hinzu, die aus den Informationsbits errechnet werden, wobei die Informationsbits selbst unverändert bleiben, wohingegen beim unsystematischen Blockcode die Informationsbits selbst verändert werden, bspw. steckt die Information in einer von einer zur nächsten Stelle durchgeführten Operation. Auch hier sind Prüfbits zur Rekonstruktion der in den Operationen versteckten Information vorgesehen. Nachfolgend wird insbesondere eine technisch bedeutende Variante unsystematischer Blockcodes, die sogenannten terminierten Faltungscodes, betrachtet.

Nun ist es entscheidend von Nachteil, eine Zuordnung des empfangenen Codeworts (mit den mit analogen Werten belegten Stellen) „hart" zu decodieren, d.h. jede Stelle dem jeweils naheliegendsten Booleschen Wert zuzuordnen, da hierbei wertvolle Information verloren geht.

Die **Aufgabe** der Erfindung besteht darin, eine Decodierung eines vorgegebenen Codeworts zu bestimmen, wobei die Decodierung analoge Werte

(sogenannte „Soft-Outputs“) liefert, die insbesondere in nachfolgenden Decodierverfahren berücksichtigt werden können und somit eine hohe Fehlerkorrektur bei der Übertragung von Codewörtern über einen gestörten Kanal ermöglichen.

Diese Aufgabe wird gemäß den Merkmalen der unabhängigen Patentansprüche gelöst. Weiterbildungen der Erfindung ergeben sich auch aus den abhängigen Ansprüchen.

Zur Lösung der Aufgabe wird ein Verfahren zur Decodierung eines vorgegebenen Codeworts angegeben, bei dem das Codewort mehrere Stellen mit unterschiedlichen Werten umfaßt. Eine Encodierung ist dabei insbesondere mit einem terminierten Faltungscode erfolgt. Jeder Stelle des Codeworts wird ein Sicherheitsmaß (Soft-Output) für einen wahrscheinlichsten Booleschen Wert zugeordnet, indem die Zuordnung basierend auf einer Trellis-Darstellung erfolgt. Durch die Zuordnung der einzelnen Stellen des Codeworts wird die Decodierung desselben bestimmt.

Hierbei ist es entscheidend von Vorteil, daß durch die auf der Trellis-Darstellung basierende Zuordnung eine deutliche Komplexitätsreduktion gegenüber einer allgemeinen Darstellung erfolgt, was dazu führt, daß auch in Echtzeit eine Decodierung des Codeworts (Erzeugung der Soft-Outputs an den Stellen des Codeworts) möglich wird.

Eine Weiterbildung besteht darin, daß die Decodiervorschrift für jede Stelle des Codeworts bestimmt ist durch

$$L(U_i|y) = \ln\left(\frac{\sum\limits_{c\in\Gamma^i(+1)} \exp\left(-\frac{(y-c)^T(y-c)}{2\sigma^2}\right)}{\sum\limits_{c\in\Gamma^i(-1)} \exp\left(-\frac{(y-c)^T(y-c)}{2\sigma^2}\right)}\right), \quad \text{für } i = 1,\ldots,K\,, \qquad (1)$$

wobei

$L(U_i\|y)$	ein Sicherheitsmaß (Soft-Output) für die i-te Stelle des zu bestimmenden Codeworts;
y	ein zu decodierendes Demodulationsergebnis;
c	ein Codewort;
$\Gamma^i(\pm 1)$	sämtliche Codewörter für $u_i = \pm 1$;
σ^2	eine Varianz (Kanalstörung)

bezeichnen.

Eine andere Weiterbildung besteht darin, daß die Gleichung (1) gelöst wird, indem eine Eigenschaft eines bei der Codierung (und entsprechend bei der Decodierung) eingesetzten Faltungscodes ausgenutzt wird, die entsprechend

einer bei der Faltung eingesetzten Schieberegisteroperation Zustände bestimmt, aus welchen Zuständen wiederum die Trellis-Darstellung hervorgeht.

Im Rahmen einer zusätzlichen Weiterbildung wird die Trellis-Darstellung in einer vorgegebenen Richtung durchlaufen, um Terme A_m bzw. $\tilde{A}_m$ rekursiv zu berechnen. In diese Berechnungsvorschrift gehen an den Knoten (s,m) der Trellis-Darstellung Knotengewichte $\mu_m(s)$ ein, die durch das Demodulationsergebnis y bestimmt werden. Die Terme A_m und $\tilde{A}_m$ werden beschrieben durch

$$\tilde{A}_m(E) = \sum_{s \in E} A_m(s), \quad \text{für } m \in \mathbb{N} \tag{2}$$

mit

$$A_m(s) = \mu_m(s) \sum_{t \in W(s,V_m)} A_{m-1}(t), \quad \text{für } m \in \mathbb{N} \tag{3}$$

und einem Startwert

$$A_0(s) = \begin{cases} 1 & : \text{ für } s = s_0, \\ 0 & : \text{ sonst.} \end{cases} \tag{4}$$

Eine detaillierte Erörterung der hier angeführten Beschreibungsformen findet sich auch in der Beschreibung des Ausführungsbeispiels.

Eine Ausgestaltung besteht darin, daß Abbildungen B_m anhand der Trellis-Darstellung ermittelt werden, wobei die Trellis-Darstellung entgegen der vorgegebenen Richtung bearbeitet wird. Der Term B_m wird bestimmt durch

$$B_m(s) = \mu_{Q-m+1}(s) \sum_{t \in T(s,V_{Q-m+2})} B_{m-1}(t), \qquad \text{für } 1 \leq m \leq Q, \tag{5}$$

wobei zur Terminierung der Rekursion

$$B_0(s) = \begin{cases} 1 & : \text{ für } s = s_0, \\ 0 & : \text{ sonst} \end{cases} \tag{6}$$

bestimmt wird.

Weiterhin können Terme A_α^i ermittelt werden, indem die Trellis-Darstellung erneut durchlaufen wird, wobei die bereits ermittelten Terme A_m und B_m berücksichtigt werden. Insbesondere werden die Terme A_α^i bestimmt gemäß

$$A_\alpha^i(y) = \sum_{s \in S} A_{j-1}(s) \sum_{t \in T(s,V_j^i(\alpha))} B_{Q-j+1}(t). \tag{7}$$

In einer weiteren Ausgestaltung werden die K Stellen des decodierten Codeworts bestimmt gemäß

$$L(U_i|y) = \ln\left(\frac{A^i_{+1}(y)}{A^i_{-1}(y)}\right), \quad i = 1, \ldots, K. \tag{8}$$

Für die Herleitung wird insbesondere ein AWGN(= **A**dditive **G**aussian **W**hite Noise)-Kanalmodell eingesetzt. Das vorgestellte Verfahren kann auch für andere Kanalmodelle, insbesondere für im Mobilfunk eingesetzte Kanalmodelle, angewandt werden.

Eine andere Ausgestaltung betrifft den Einsatz des Verfahrens in einem Mobilfunknetz, insbesondere dem GSM-Netz.

Auch ist es eine Weiterbildung, daß nach der Ermittlung der Soft-Outputs eine „harte" Zuordnung der analogen Werte zu den Booleschen Werten ± 1 erfolgt. Dabei wird jeweils der nächstliegende Boolesche Wert für die Zuordnung des analogen Werts ermittelt.

Die ermittelten Soft-Output-Werte können als Eingabewerte einer weiteren Decodierung bei Verwendung verketteter Codes dienen.

Zur Lösung der Aufgabe wird ferner eine Anordnung zur Decodierung eines vorgegebenen Codeworts angegeben, bei der eine Prozessoreinheit vorgesehen ist, die derart eingerichtet ist, daß

1. das Codewort mehrere Stellen mit unterschiedlichen Werten umfaßt;
2. jede Stelle des Codeworts einem Soft-Output-Wert zuordenbar ist, indem die Zuordnung basierend auf einer Trellis-Darstellung erfolgt;
3. durch die Zuordnung der einzelnen Stellen des Codeworts die Decodierung desselben bestimmbar ist.

Diese Anordnung ist insbesondere geeignet zur Durchführung des erfindungsgemäßen Verfahrens oder einer seiner vorstehend erläuterten Weiterbildungen.

Ausführungsbeispiele der Erfindung werden nachfolgend anhand der Zeichnung dargestellt und erläutert.

Es zeigen

Fig.1 eine Darstellung zur digitalen Nachrichtenübertragung;

Fig.2 einen Algorithmus in Pseudocode-Notation zum Fortschreiten im Trellis-Diagramm unter Betrachtung aller Zustände zur Berechnung von Knotengewichten;

Fig.3 einen Algorithmus in Pseudeocode-Noation zur Ermittlung von Soft-Outputs (allgemeiner Fall);

Fig.4 einen Algorithmus in Pseudeocode-Noation zur Ermittlung von Soft-Outputs (Spezialfall: binärer Zustandsübergang);

Fig.5 eine Prozessoreinheit.

Nachfolgend werden zunächst der Faltungscode, dann die Komplexitätsreduktion bei der Berechnung von Soft-Outputs und schließlich eine algorithmische Umsetzung der Komplexitätsreduktion näher beschrieben.

Terminierter Faltungscode

In der Nachrichtentechnik werden terminierte Faltungscodes meist in Verkettung mit weiteren systematischen oder unsystematischen Blockcodes eingesetzt. Insbesondere dient dabei das Decodierungsergebnis eines Faltungsdecoders als Eingabe für einen weiteren Decoder.

Um eine möglichst niedrige Fehlerrate zu gewährleisten, ist es nötig, „weiche“ statt „harte“ Decodierungsentscheidungen bei der Faltungsdecodierung für den weiteren Decoder zu liefern, d.h. ein Tupel von „weichen“ Werten (Soft-Outputs) aus $\mathbb{R}$ zu erzeugen anstatt ein Tupel von „harten“ Booleschen (± 1) Werten. Der Betrag der jeweiligen „weichen“ Entscheidung gibt dann ein Sicherheitsmaß für die Richtigkeit der Entscheidung an.

Abhängig vom Kanalmodell ist eine Berechnung dieser Soft-Outputs nach Gleichung (1) prinzipiell möglich. Allerdings beträgt dabei die numerische Komplexität zur Berechnung eines Soft-Outputs $O(2^K)$, wobei K die Zahl der Nachrichtenbits angibt. Bei realistisch großen K sind diese Formeln also nicht auswertbar, insbesondere, da alle paar Millisekunden erneut ein solches Codewort zu berechnen ist (Echtzeitanforderung).

Eine Konsequenz lag in dem Verzicht auf Soft-Outputs (mit allen Konsequenzen für die Wort- und Bitfehlerraten) bzw. es wurden weniger aufwendige Approximationen zur Bestimmung der Soft-Outputs durchgeführt.

Im folgenden wird eine Möglichkeit für terminierte Faltungscodes angegeben, mit deren Hilfe in einer Trellis-Darstellung diese Komplexität auf $O(K)$ zur Berechnung aller Soft-Outputs reduziert werden kann, d.h. eine exakte Auswertung der Gleichung (1) wird damit möglich.

Nachfolgend werden die Bits des Codes in $\{\pm 1\}$-Repräsentation dargestellt. Im Vergleich zu einer informationstechnisch oft üblichen $\{0, 1\}$-Repräsentation korrespondiert -1 mit 1 und 1 mit 0.

Auf einem Körper $\{\pm 1\}$ sind Addition $\oplus$ und Multiplikation $\odot$ wie folgt definiert:

$$\begin{aligned} -1 \oplus -1 &= 1 \\ -1 \oplus 1 &= -1 \\ 1 \oplus -1 &= -1 \\ 1 \oplus 1 &= 1 \end{aligned} \qquad \begin{aligned} -1 \odot -1 &= -1 \\ -1 \odot 1 &= 1 \\ 1 \odot -1 &= 1 \\ 1 \odot 1 &= 1 \end{aligned}$$

Die Codierung erfolgt mit Hilfe eines „Schieberegisters", in welches taktweise Bitblöcke (Eingabeblöcke) der Nachrichtenbits (Informationsbits) geschrieben werden. Die Kombination der Bits des Schieberegisters erzeugt dann einen Bitblock des Codeworts. Das Schieberegisters ist je mit $+1$ Bits vorbelegt. Zum Abschluß der Codierung (Terminierung) werden Blöcke von Tail-Nullen $(+1)$ nachgeschoben. Wie eingangs erwähnt wurde, werden mittels Codierung den Informationsbits Prüfbits zugeordnet, anhand derer Bitfehler korrigiert werden können.

Für die weiteren Ausführungen werden definiert:

$b \in \mathbb{N}$	Anzahl der Eingabebits pro Block
$V := \{\pm 1\}^b$	Menge der Zustandsübergangszeichen
$a \in \mathbb{N}$	Anzahl der Eingabeblöcke
$K := a \cdot b$	Anzahl der Nachrichtenbits ohne Tail-Nullen
$k \in \mathbb{N},\ k \geq 2$	Blocklänge des Schieberegisters, Eindringtiefe
$L := k \cdot b$	Bitlänge des Schieberegisters
$S := \{\pm 1\}^L$	Menge der Schieberegisterzeichen
$n \in \mathbb{N}$	Anzahl der Ausgabebits pro Block
$Q := a + k - 1$	Anzahl der Zustandsübergänge, Eingabeblöcke + Nullen
$N := n \cdot Q$	Anzahl der Codebits
$R := \frac{b}{n}$	Coderate

Hierbei sei angemerkt, daß die Coderate nicht K/N beträgt, da die Nachrichtenbits ohne die Nullen (+1) der Faltungsterminierung gezählt wurden.

Weiterhin seien $s_0 \in S$ und $v_0 \in V$ die jeweiligen Nullelemente, d.h.

$$s_0 = (+1, \ldots, +1)^\mathsf{T}, \quad v_0 = (+1, \ldots, +1)^\mathsf{T}. \tag{9}$$

Die Zustandsübergangsfunktion des Schieberegisters sei

$$T: \; S \times V \to S, \tag{10}$$

$$(s, v) \mapsto (s^{b+1}, \ldots, s^L, v^1, \ldots, v^b)^\mathsf{T}. \tag{11}$$

Der terminierte Faltungscode wird über die charakterisierenden Teilmengen

$$M_1, \ldots, M_n \subseteq \{1, \ldots, L\}, \tag{12}$$

definiert (Kombination der Registerbits, alternativ in Polynomdarstellung).

Die Codierung des aktuellen Registerinhaltes erfolgt über

$$C: \; S \to \{\pm 1\}^n, \tag{13}$$

$$s \mapsto C(s) \quad \text{mit } C_j(s) := \bigoplus_{i \in M_j} s^i, \quad \text{für } 1 \le j \le n. \tag{14}$$

Dabei steht s^i für die i-te Komponente von s.

Die Codierung eines Nachrichtenwortes ist schließlich definiert mittels

$$\varphi: \; \{\pm 1\}^K \to \{\pm 1\}^N, \tag{15}$$

$$u \mapsto \begin{pmatrix} C(s_1) \\ \vdots \\ (s_Q) \end{pmatrix}, \tag{16}$$

wobei $s_0 \in S$ der Nullzustand (Nullelement) ist,

$$u = \begin{pmatrix} \nu_1 \\ \vdots \\ \nu_a \end{pmatrix}, \quad \nu_i \in V, \quad 1 \le i \le a, \tag{17}$$

$$\nu_i := v_0, \quad a+1 \le i \le Q, \tag{18}$$

und weiter

$$s_i := T(s_{i-1}, \nu_i), \quad 1 \le i \le Q. \tag{19}$$

Nach Definition von T ergibt sich

$$s_{Q+1} := T(s_Q, v_0) = s_0. \tag{20}$$

Die Menge aller Codeworte ist demnach

$$\varphi\left(\{\pm 1\}^K\right) := \left\{\varphi(u) \in \{\pm 1\}^N;\ u \in \{\pm 1\}^K\right\}. \tag{21}$$

Oft werden anstatt der Mengen M_j Polynome

$$p_j \in \{0,1\}[D] \quad \text{mit} \quad \deg(p_j) \leq L-1$$

zur Codedefinition verwendet, d.h.

$$p_j(D) = \sum_{i=0}^{L-1} \gamma_{i,j} D^i, \tag{22}$$

$$\text{mit } \gamma_{i,j} \in \{0,1\} \qquad \begin{array}{l} i = 0, \ldots, L-1, \\ j = 1, \ldots, n. \end{array}$$

Es gelten dann für $j = 1, \ldots, n$ die Umformungen:

$$M_j = \{i \in \{1, \ldots, L\};\ \gamma_{L-i,j} = 1\} \tag{23}$$

$$p_j(D) = \sum_{i \in M_j} D^{L-i}. \tag{24}$$

Blockcode-Darstellung

Da ein terminierter Faltungscode ein Blockcode ist, lassen sich die Codebits c_j, $1 \leq j \leq N$, aus den Nachrichtenbits u_i, $1 \leq i \leq K$, mit Indexmengen J_j auch wie folgt darstellen:

$$c_j := \bigoplus_{i \in J_j} u_i, \quad \text{für } 1 \leq j \leq N, \tag{25}$$

wobei

$$J_1, \ldots, J_N \subseteq \{1, \ldots, K\}. \tag{26}$$

gilt. Die Indexmengen J_j lassen sich direkt aus den obigen Indexmengen M_m der Codedefinition berechnen.

Betrachte

$$j = n(q-1)+m, \quad q = 1,\dots,Q,\ m = 1,\dots,n. \tag{27}$$

$$c_j = C_m(s_q) = \bigoplus_{i\in M_m} (s_q)^i = \bigoplus_{i\in M_m} u_{i+b(q-k)}, \tag{28}$$

wobei $u_i := +1$ für $i \notin \{1,\dots,K\}$ gilt.

Ferner gilt

$$c_j = \bigoplus_{i-b(q-k)\in M_m} u_i = \bigoplus_{i\in M_m+b(q-k)} u_i, \tag{29}$$

und somit folgt für $j = 1,\dots,N$

$$\begin{aligned} J_j &= \{1,\dots,K\} \cap (M_m + b(q-k)) \\ &= \{i \in \{1,\dots,K\};\ i - b(q-k) \in M_m\}\,. \end{aligned} \tag{30}$$

Beispiel: SACCH–Faltungscode

Der in der GSM Technical Specification GSM 05.03, Version 5.2.0 (Channel coding), im Abschnitt 4.1.3 beschriebene Faltungscode lautet in obiger Terminologie:

$b = 1$	Anzahl der Eingabebits pro Block
$V = \{\pm 1\}$	Menge der Zustandsübergangszeichen
$a = 224$	Anzahl der Eingabeblöcke
$K = 224$	Anzahl der Nachrichtenbits ohne Tail-Nullen
$k = 5$	Blocklänge des Schieberegisters, Eindringtiefe
$L = 5$	Bitlänge des Schieberegisters
$S = \{\pm 1\}^5$	Menge der Schieberegisterzeichen
$n = 2$	Anzahl der Ausgabebits pro Block
$Q = 228$	Anzahl der Zustandsübergänge, Eingabeblöcke + Nullen
$N = 456$	Anzahl der Codebits
$R = \frac{1}{2}$	Coderate
$M_1 = \{1,2,5\}$	charakterisierende Menge; Polynom: $1 + D^3 + D^4$
$M_2 = \{1,2,4,5\}$	charakterisierende Menge; Polynom: $1 + D + D^3 + D^4$

Soft-Outputs bei einem AWGN-Kanalmodell

Nachfolgend werden insbesondere der Übersicht halber Berechnungsvorschriften zur Ermittlung der Soft-Outputs hergeleitet.

Dazu werden ein Wahrscheinlichkeitsraum $(\Omega, \mathcal{S}, P)$ und eine K-dimensionale Zufallsvariable $U : \Omega \to \{\pm 1\}^K$ mit den Eigenschaften

- Die Komponenten $U_1, \ldots, U_K : \Omega \to \{\pm 1\}$ sind stochastisch unabhängig.
- Für $i = 1, \ldots, K$ gilt

$$P(\{\omega \in \Omega;\ U_i(\omega) = -1\}) = P(\{\omega \in \Omega;\ U_i(\omega) = +1\}). \tag{31}$$

betrachtet.

Fig.1 zeigt eine Darstellung zur digitalen Nachrichtenübertragung. Eine Einheit aus Quelle 201, Quellencodierer 202 und Kryptocodierer 203 bestimmt eine Information $u \in \{\pm 1\}^K$, die als Eingabe für einen (ggf. auch mehrere) Kanalcodierer 204 dient. Der Kanalcodierer 204 erzeugt ein Codewort $c \in \{\pm 1\}^N$, das in einen Modulator 205 eingespeist und über einen gestörten physikalischen Kanal 206 zu einem Empfänger übertragen wird, wo es in einem Demodulator 207 zu einem reellwertigen Codewort $y \in \mathbb{R}^N$ bestimmt wird. Dieses Codewort wird in einem Kanaldecodierer 208 in eine reellwertiges Information umgesetzt. Gegebenenfalls kann in einem weiteren Decodierer auch eine „harte" Zuordnung zu den Booleschen Werten ± 1 getroffen werden, so daß die empfangene Information in Boolescher Notation vorliegt. Eine Einheit aus Kryptodecoder 209, Quellendecoder 210 und Senke 211 komplettiert den Empfänger. Die beiden Einheiten Kryptocodierer 203 und Kryptodecoder 209 sind dabei optional.

Die zu rekonstruierende Information $u \in \{\pm 1\}^K$ des Kryptocodierers 203 wird als Realisierung der Zufallsvariablen U interpretiert, da beim Empfänger nichts über die Wahl von u bekannt ist.

Die Ausgabe $c \in \{\pm 1\}^N$ des Kanalcodierers 204 ist also eine Realisierung der Zufallsvariablen $\varphi(U)$.

Die Ausgabe $y \in \mathbb{R}^N$ des Demodulators 207 wird als Realisierung der Zufallsvariablen

$$Y : \Omega \to \mathbb{R}^N, \tag{32}$$

$$\omega \mapsto \varphi(U(\omega)) + Z(\omega), \tag{33}$$

interpretiert, wobei eine Zufallsvariable $Z : \Omega \to \mathbb{R}^N$ die Kanalstörung auf dem physikalischen Kanal 206 repräsentiert.

Im folgenden wird ein AWGN-Kanalmodell angenommen, d.h. Z ist eine $\mathcal{N}(0, \sigma^2 I_N)$ normalverteilte Zufallsvariable, die stochastisch unabhängig von U bzw. $\varphi(U)$ ist. Die Varianz σ^2 berechnet sich aus dem Verhältnis von Rauschleistungsdichte und mittlerer Energie auf dem Kanal 206 und wird hier als bekannt vorausgesetzt.

Basierend auf einer Realisierung y von Y soll die unbekannte Ausgabe $u \in \{\pm 1\}^K$ des Kryptocodierers rekonstruiert werden. Um die unbekannten Größen $u_1, \ldots, u_K$ zu schätzen, wird die Verteilung der Zufallsvariablen U unter der Bedingung, daß y empfangen wurde, untersucht.

Die Tatsache, daß die Zufallsvariable Y eine stetige Zufallsgröße ist, hat zur Folge, daß die Betrachtung von U unter der Bedingung, daß y empfangen wurde ($Y(\hat{\omega}) = y$), äußerst kompliziert ist.

Zunächst wird für $i \in \{1, \ldots, K\}$ und $\alpha \in \{\pm 1\}$ definiert

$$\Gamma^i(\alpha) := \{\varphi(u);\ u \in \{\pm 1\}^K;\ u_i = \alpha\}\,. \tag{34}$$

In einem vorbereitenden Schritt werden für $\epsilon > 0$ und unter Beachtung der Injektivität der Codierungsabbildung φ die folgenden Größen betrachtet:

$$\begin{aligned} L_\epsilon(U_i|y) &:= \ln\left(\frac{P(\{\omega \in \Omega;\ U_i(\omega) = +1\} \mid \{\omega \in \Omega;\ Y(\omega) \in M_{y,\epsilon}\})}{P(\{\omega \in \Omega;\ U_i(\omega) = -1\} \mid \{\omega \in \Omega;\ Y(\omega) \in M_{y,\epsilon}\})}\right) \\ &= \ln\left(\frac{\sum\limits_{c \in \Gamma^i(+1)} P(\{\omega \in \Omega;\ \varphi(U(\omega)) = c\} \mid \{\omega \in \Omega;\ Y(\omega) \in M_{y,\epsilon}\})}{\sum\limits_{c \in \Gamma^i(-1)} P(\{\omega \in \Omega;\ \varphi(U(\omega)) = c\} \mid \{\omega \in \Omega;\ Y(\omega) \in M_{y,\epsilon}\})}\right), \end{aligned} \tag{35}$$

für $i = 1, \ldots, K$, wobei $M_{y,\epsilon} := [y_1, y_1 + \epsilon] \times \ldots \times [y_N, y_N + \epsilon]$ gilt.

Durch Anwendung des Satzes von Bayes ergibt sich:

$$\begin{aligned} L_\epsilon(U_i|y) &= \ln\left(\frac{\sum\limits_{c \in \Gamma^i(+1)} P(\{\omega \in \Omega;\ Y(\omega) \in M_{y,\epsilon}\} \mid \{\omega \in \Omega;\ \varphi(U(\omega)) = c\})}{\sum\limits_{c \in \Gamma^i(-1)} P(\{\omega \in \Omega;\ Y(\omega) \in M_{y,\epsilon}\} \mid \{\omega \in \Omega;\ \varphi(U(\omega)) = c\})}\right) \\ &= \ln\left(\frac{\sum\limits_{c \in \Gamma^i(+1)} \int\limits_{M_{y,\epsilon}} \exp\left(-\frac{(x-c)^T(x-c)}{2\sigma^2}\right) dx}{\sum\limits_{c \in \Gamma^i(-1)} \int\limits_{M_{y,\epsilon}} \exp\left(-\frac{(x-c)^T(x-c)}{2\sigma^2}\right) dx}\right). \end{aligned} \tag{36}$$

Wird nun durch mehrfache Verwendung der Regel von L'Hospital der Grenzübergang von $L_\epsilon(U_i|y)$ für $\epsilon \downarrow 0$ betrachtet, so erhält man für jedes Zeichen den Soft-Output $L(U_i|y)$ wie in Gleichung (1).

Da

$$\Gamma^i(+1) \cup \Gamma^i(-1) = \{\pm 1\}^K$$

gilt, sind zur Auswertung von Gleichung (1) insgesamt $O(2^K)$ numerische Operationen notwendig.

Der Vektor $L(U_\bullet|y) \in \mathbb{R}^K$ ist das Ergebnis des Decodierers 208.

Komplexitätsreduktion bei der Bestimmung der Soft-Outputs

Soft-Output–Bestimmung für Faltungscodes

Zunächst werden die speziellen Eigenschaften der terminierten Faltungscodierung zu einer aufgegliederten Darstellung der Soft-Output-Formel (1) eingesetzt.

Es wird zu einer beliebigen aber fest gewählten Ausgabe $y \in \mathbb{R}^N$ des Demodulators 207 die folgende Bewertungsfunktion (eine Viterbi-Metrik) von Codewörtern betrachtet:

$$F: \{\pm 1\}^N \to \mathbb{R}_0^+, \tag{37}$$

$$c \mapsto \sum_{j=1}^{N} (y_j - c_j)^2. \tag{38}$$

Für zulässige Codewörter $c \in \{\pm 1\}^N$, d.h., $c \in \varphi(\{\pm 1\}^K)$, läßt sich $F(c)$ mit der Schieberegister-Darstellung wie folgt zerlegen:

$$F(c) = \sum_{q=1}^{Q} \underbrace{\sum_{j=1}^{n} \left(y_{n(q-1)+j} - C_j(\tilde{s}_q^c)\right)^2}_{=:\Delta F_q(\tilde{s}_q^c)}, \tag{39}$$

wobei $\tilde{s}_q^c$ für den q-ten Zustand des Schieberegisters bei der (eindeutigen) Erzeugung des Wortes c steht.

Nun wird für $i = 1, \ldots, K$ und $\alpha \in \{\pm 1\}$ definiert:

$$A_\alpha^i(y) := \sum_{c \in \Gamma^i(\alpha)} \exp\left(-\frac{(y-c)^\top (y-c)}{2\sigma^2}\right) = \sum_{c \in \Gamma^i(\alpha)} \prod_{q=1}^{Q} \exp\left(-\frac{1}{2\sigma^2} \Delta F_q(\tilde{s}_q^c)\right). \tag{40}$$

Damit gilt also für die Soft-Outputs

$$L(U_i|y) = \ln\left(\frac{A_{+1}^i(y)}{A_{-1}^i(y)}\right), \quad i = 1, \ldots, K. \tag{41}$$

Nachfolgend werden die Werte $A_\alpha^i(y)$ mit Hilfe einer Trellis-Diagramm-Darstellung (auch: Trellis-Diagramm oder Trellis-Darstellung) bestimmt.

Zur Reduktion der Berechnungskomplexität wird in den folgenden Abschnitten wie folgt vorgegangen:

- Verallgemeinerung von A_α^i durch Abbildungen $\tilde{A}_m$.
- Rekursive Darstellung von $\tilde{A}_m$ durch Abbildungen A_m, deren Werte mit einem „Von-Links-nach-Rechts"-Durchlauf eines Trellis-Diagramms berechnet werden.
- Umkehrung der Rekursion durch Abbildungen B_m, deren Werte mit einem „Von-Rechts-nach-Links"-Durchlauf eines Trellis-Diagramms berechnet werden.
- Gemeinsame Berechnung aller A_α^i mittels eines weiteren Trellis-Diagramm-Durchlaufs unter Verwendung von A_m und B_m.

Als Trellis-Diagramm wird hier eine Menge

$$\mathcal{T} = \{(s, q);\ s \in S,\ q = 0, \ldots, Q+1\} \tag{42}$$

benannt. Die Elemente (s, q) dieser Menge werden auch als Knoten im Trellis-Diagramm bezeichnet, wobei s einen Zustand darstellt und q als dynamischer Wert (insbesondere die Zeit) angesehen wird.

Allgemeine rekursive Darstellung

Zunächst sind einige Definitionen nötig, um die A_α^i in einer verallgemeinerten Form darzustellen, die eine spätere Umformung erlaubt. Deshalb wird bestimmt

$$s_1^u := T(s_0, u_1), \quad u \in V^m = V \times \ldots \times V, \; m \geq 1, \tag{43}$$

$$s_j^u := T(s_{j-1}^u, u_j) \quad u \in V^m, \; m \geq j > 2, \tag{44}$$

d.h., s_j^u repräsentiert den Zustand des Schieberegisters nach j Shifts des Registers mit den Eingabezeichen $u_1, \ldots, u_j$.

Weiterhin werden Mengen $V_j \subseteq V$, $j \in \mathbb{N}$, die die zulässigen Zustandsübergangszeichen im j-ten Schritt enthalten, betrachtet. Ferner werden Produktmengen definiert zu

$$U_m := V_1 \times \ldots \times V_m \subseteq V^m, \quad m \in \mathbb{N}, \tag{45}$$

d.h., U_m enthält die ersten m Komponenten der zulässigen Eingabeworte.

Für $q \in \mathbb{N}$ werden Abbildungen

$$\mu_q : S \to \mathbb{R} \tag{46}$$

betrachtet, und für $m \in \mathbb{N}$ und Eingabewortmengen $U_m \subseteq V^m$ werden Abbildungen definiert

$$\tilde{A}_m : \wp(S) \to \mathbb{R}, \tag{47}$$

$$E \mapsto \sum_{\substack{(u \in U_m) \\ \wedge (s_m^u \in E)}} \prod_{j=1}^{m} \mu_j(s_j^u), \tag{48}$$

d.h. es wird über alle zulässigen Eingabeworte summiert, deren Schieberegister einen Endzustand in E erreicht. Falls es keine solchen Eingabeworte gibt, so ist die Summe über eine leere Indexmenge zu 0 bestimmt.

Zusätzlich wird eine Abbildung bestimmt zu

$$W : S \times \wp(V) \to \wp(S), \tag{49}$$

$$(t, \hat{V}) \mapsto \left\{ s \in S; \; \exists \hat{v} \in \hat{V} \ni T(s, \hat{v}) = t \right\}, \tag{50}$$

d.h., W bildet $(t, \hat{V})$ in die Menge aller Zustände ab, die den Zustand t mit einem Übergangszeichen aus $\hat{V}$ erreichen können.

Es gilt für $m \geq 2$, $E \subseteq S$

$$\begin{aligned}
\tilde{A}_m(E) &= \sum_{\substack{(u \in U_m) \\ \wedge (s_m^u \in E)}} \prod_{j=1}^{m} \mu_j(s_j^u) \\
&= \sum_{s \in E} \sum_{\substack{(u \in U_m) \\ \wedge (s_m^u = s)}} \prod_{j=1}^{m} \mu_j(s_j^u) \\
&= \sum_{s \in E} \mu_m(s) \sum_{\substack{(u \in U_m) \\ \wedge (s_m^u = s)}} \prod_{j=1}^{m-1} \mu_j(s_j^u) \\
&= \sum_{s \in E} \mu_m(s) \sum_{\substack{(u \in U_{m-1}) \\ \wedge (s_{m-1}^u \in W(s, V_m))}} \prod_{j=1}^{m-1} \mu_j(s_j^u) \\
&= \sum_{s \in E} \mu_m(s) \tilde{A}_{m-1}\left(W(s, V_m)\right). \qquad (51)
\end{aligned}$$

Bei der Umformung im vorletzten Schritt ist zu beachten, daß es **genau ein** Übergangszeichen $v \in V_m$ gibt mit $T(s_{m-1}^u, v) = s$, wenn s_{m-1}^u in $W(s, V_m)$ liegt, d.h., es sind keine Vielfachheiten zu beachten.

Nun betrachte man für $m \geq 2$ Abbildungen

$$A_m : S \to \mathbb{R}, \qquad (52)$$

$$s \mapsto \mu_m(s) \tilde{A}_{m-1}\left(W(s, V_m)\right). \qquad (53)$$

Somit läßt sich für $m \geq 3$ eine Rekursionsformel ableiten:

$$\begin{aligned}
A_m(s) &= \mu_m(s) \tilde{A}_{m-1}\left(W(s, V_m)\right) \\
&= \mu_m(s) \sum_{t \in W(s, V_m)} \mu_{m-1}(t) \tilde{A}_{m-2}\left(W(t, V_{m-1})\right) \\
&= \mu_m(s) \sum_{t \in W(s, V_m)} A_{m-1}(t). \qquad (54)
\end{aligned}$$

Weiter gilt:

$$\begin{aligned} A_2(s) &= \mu_2(s)\tilde{A}_1\left(W(s,V_2)\right) \\ &= \mu_2(s) \sum_{\substack{(u\in U_1)\\ \wedge(s_1^u\in W(s,V_2))}} \mu_1(s_1^u) \\ &= \mu_2(s) \sum_{t\in W(s,V_2)} \mu_1(t)\delta_{s_0\in W(t,V_1)} \\ &= \mu_2(s) \sum_{t\in W(s,V_2)} \underbrace{\mu_1(t) \sum_{\hat{t}\in W(t,V_1)} \underbrace{\delta_{\hat{t}=s_0}}_{=:A_0(\hat{t})}}_{=:A_1(t)} \end{aligned} \tag{55}$$

Zusammenfassend gilt also für $s \in S$, $E \subseteq S$:

$$A_0(s) = \begin{cases} 1, & \text{für } s = s_0, \\ 0, & \text{sonst} \end{cases}, \tag{56}$$

$$A_m(s) = \mu_m(s) \sum_{t\in W(s,V_m)} A_{m-1}(t), \quad \text{für } m \in \mathbb{N}, \tag{57}$$

$$\tilde{A}_m(E) = \sum_{s\in E} A_m(s), \quad \text{für } m \in \mathbb{N}. \tag{58}$$

Die Mengen $W(s,V_m)$ können konstruktiv dargestellt werden. Dazu werden zwei weitere Abbildungen betrachtet. Es wird definiert

$$\tau : S \to V, \tag{59}$$

$$s = (s^1,\ldots,s^L)^\top \mapsto (s^{L-b+1},\ldots,s^L)^\top, \tag{60}$$

d.h., wenn der Zustand s Ergebnis eines Zustandsübergangs ist, so war $\tau(s)$ das zugehörige Zustandsübergangszeichen.

Weiter wird definiert

$$\hat{T}:\ V \times S \to S, \tag{61}$$

$$(v,s) \mapsto (v^1,\ldots,v^b,s^1,\ldots,s^{L-b})^\top, \tag{62}$$

d.h. $\hat{T}$ dreht die Richtung der Schieberegisteroperation um.

Es gilt dann

$$T\left(\hat{T}(v,s),\tau(s)\right) = s, \quad \text{für alle } s \in S,\ v \in V \tag{63}$$

und für alle $t \in S$ und $\hat{V} \subseteq V$ gilt ferner

$$W(t,\hat{V}) = \left\{ s \in S;\ \exists \hat{v} \in \hat{V} \ni T(s,\hat{v}) = t \right\} = \begin{cases} \left\{ \hat{T}(v,t);\ v \in V \right\}, & \text{falls } \tau(t) \in \hat{V}, \\ \emptyset, & \text{sonst.} \end{cases} \tag{64}$$

Die Rekursionsformel (57) für $A_m(s)$ läßt sich also wie folgt konstruktiv aufschreiben:

$$A_m(s) = \mu_m(s) \sum_{t \in W(s,V_m)} A_{m-1}(t) = \begin{cases} \mu_m(s) \sum\limits_{v \in V} A_{m-1}\left(\hat{T}(v,s)\right), & \text{falls } \tau(s) \in V_m, \\ 0, & \text{sonst.} \end{cases} \tag{65}$$

Es sei vermerkt, daß in diesem Abschnitt keinerlei Einschränkungen an die Menge V der Zustandsübergangszeichen und an die Mengen $V_j \in \wp(V)$ gemacht wurden.

Rekursionsumkehrung

Im folgenden wird eine Rekursion in „umgekehrter Richtung“ gegenüber der obigen Rekursion beschrieben. Diese neue Rekursion wird mit Hilfe der Rekursionsformel (57) für $A_m(s)$ definiert.

Dazu sei

$$T(t,\hat{V}) := \left\{ T(t,\hat{v});\ \hat{v} \in \hat{V} \right\}, \quad \text{für } t \in S,\ \hat{V} \subseteq V \tag{66}$$

und für $M \in \mathbb{N}$, $0 \leq m \leq Q$ betrachtet man Abbildungen

$$B_m : S \to \mathbb{R}, \tag{67}$$

mit folgender rekursiver Eigenschaft:

$$\begin{aligned}
&\sum_{s\in S} A_m(s) \sum_{t\in T(s,V_{m+1})} B_{Q-m}(t) = \\
&= \sum_{s\in S} \mu_m(s) \sum_{\hat{t}\in W(s,V_m)} A_{m-1}(\hat{t}) \sum_{t\in T(s,V_{m+1})} B_{Q-m}(t) \\
&= \sum_{\hat{t}\in S} \sum_{s\in T(\hat{t},V_m)} \mu_m(s) A_{m-1}(\hat{t}) \sum_{t\in T(s,V_{m+1})} B_{Q-m}(t) \\
&= \sum_{\hat{t}\in S} A_{m-1}(\hat{t}) \sum_{s\in T(\hat{t},V_m)} \underbrace{\mu_m(s) \sum_{t\in T(s,V_{m+1})} B_{Q-m}(t)}_{=:B_{Q-m+1}(s)},
\end{aligned}$$

d.h.

$$\sum_{s\in S} A_m(s) \sum_{t\in T(s,V_{m+1})} B_{Q-m}(t) = \sum_{s\in S} A_{m-1}(s) \sum_{t\in T(s,V_m)} B_{Q-m+1}(t). \qquad (68)$$

Durch mehrfache Anwendung der Gleichung (68) ergibt sich für ein beliebiges $j \in \{1,\dots,m+1\}$

$$\sum_{s\in S} A_m(s) \sum_{t\in T(s,V_{m+1})} B_{Q-m}(t) = \sum_{s\in S} A_{j-1}(s) \sum_{t\in T(s,V_j)} B_{Q-j+1}(t). \qquad (69)$$

Nach obiger Definition lautet also die Rekursionsformel

$$B_m(s) = \mu_{Q-m+1}(s) \sum_{t\in T(s,V_{Q-m+2})} B_{m-1}(t), \qquad \text{für } 1 \le m \le Q. \qquad (70)$$

Zur Terminierung der Rekursion werden definiert

$$B_0(s) = \begin{cases} 1, & \text{für } s = s_0, \\ 0, & \text{sonst} \end{cases}. \qquad (71)$$

Mit dieser Terminierung und den Gleichungen (58) sowie (69) läßt sich

$$\tilde{A}_Q(W(s_0, V_{Q+1}))$$

für $V_{Q+1} := \{v_0\}$ und mit einem beliebigen $j \in \{1,\dots,Q+1\}$ wie folgt

darstellen

$$\begin{aligned}\tilde{A}_Q(W(s_0, V_{Q+1})) &= \sum_{s \in W(s_0, V_{Q+1})} A_Q(s) \\ &= \sum_{s \in S} A_Q(s) \sum_{t \in T(s, \{v_0\})} B_0(t) \\ &= \sum_{s \in S} A_Q(s) \sum_{t \in T(s, V_{Q+1})} B_0(t) \\ &= \sum_{s \in S} A_{j-1}(s) \sum_{t \in T(s, V_j)} B_{Q-j+1}(t). \end{aligned} \tag{72}$$

Bemerkung: Bei der Auswertung von (72) geht V_j nicht in die Berechnung der benötigten A_m bzw. B_m ein.

Berechnung von A_α^i

Mit den Vorarbeiten aus den vorangegangenen Abschnitten läßt sich A_α^i nun auf einfache Weise berechnen.

Hierzu werden definiert:

$$V_j := V, \quad \text{für } j \in \{1, \ldots, a\}, \tag{73}$$

$$V_j := \{v_0\}, \quad \text{für } j \in \{a+1, \ldots, Q+1\}, \tag{74}$$

d.h. alle zulässigen Codeworte sind über die Zustände s_j^u mit

$$u \in U_Q = V_1 \times \ldots \times V_Q$$

definiert.

Die bei der Berechnung der A_α^i verwendeten Codeworte sind durch $u_i = \alpha$ eingeschränkt. Zu einer beliebigen aber festen Wahl von $i \in \{1, \ldots, K\}$ gibt es genau ein $j \in \{1, \ldots, a\}$ und genau ein $\hat{i} \in \{1, \ldots, n\}$ mit

$$i = (j-1) \cdot n + \hat{i}. \tag{75}$$

Ferner werden für eine beliebige aber feste Wahl von $\alpha \in \{\pm 1\}$ definiert:

$$V_j^{\hat{i}}(\alpha) := \{v \in V;\ v_{\hat{i}} = \alpha\} \tag{76}$$

$$U_Q^{\hat{i}}(\alpha) := V_1 \times \ldots \times V_{j-1} \times V_j^{\hat{i}}(\alpha) \times V_{j+1} \times \ldots \times V_Q \subset U_Q, \tag{77}$$

d.h., die Codeworte aus $\Gamma^i(\alpha)$ sind über die Zustände s_j^u mit $u \in U_Q^i(\alpha)$ bestimmt.

Zu einer beliebigen aber festen Wahl von $y \in \mathbb{R}^N$ definiere für $q \in \{1, \ldots, Q\}$

$$\mu_q : S \to \mathbb{R}, \tag{78}$$

$$s \mapsto \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{n}\left(y_{n(q-1)+j} - C_j(s)\right)^2\right) = \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(s)\right). \tag{79}$$

Nach Definition des Faltungscodes gilt für alle s_Q^u mit $u \in U_Q$

$$s_{Q+1}^u = T(s_Q^u, u_{Q+1}) = s_0, \quad u_{Q+1} \in V_{Q+1} = \{v_0\}, \tag{80}$$

also

$$s_Q^u \in W(s_0, V_{Q+1}). \tag{81}$$

Damit gilt unter Beachtung von Gleichung (72)

$$\begin{aligned} A_\alpha^i(y) &= \sum_{c\in\Gamma^i(\alpha)}\prod_{q=1}^{Q}\exp\left(-\frac{1}{2\sigma^2}\Delta F_q(\tilde{s}_q^c)\right) \\ &= \sum_{u\in U_Q^i(\alpha)}\prod_{q=1}^{Q}\mu_q(s_q^u) \\ &= \sum_{\substack{(u\in U_Q^i(\alpha))\\ \wedge(s_Q^u\in W(s_0,V_{Q+1}))}}\prod_{q=1}^{Q}\mu_q(s_q^u) \\ &= \tilde{A}_Q(W(s_0, V_{Q+1})) \\ &= \sum_{s\in S} A_{j-1}(s)\sum_{t\in T(s,V_j^i(\alpha))} B_{Q-j+1}(t) \end{aligned} \tag{82}$$

Wichtig ist, daß die benötigten A_m und B_m unabhängig von i und α über U_Q bzw. U_{Q+1} berechnet werden können. Oben war formal $\tilde{A}_Q(W(s_0, V_{Q+1}))$ über das Hilfskonstrukt $U_Q^i(\alpha)$ bestimmt, welches in der resultierenden expliziten Darstellung aber nicht mehr benötigt wird.

Zusammenfassung der Vorgehensweise:

- Definiere

$$\begin{aligned} V_j &:= V, && \text{für } j \in \{1,\dots,a\},\\ V_j &:= \{v_0\}, && \text{für } j \in \{a+1,\dots,Q+1\},\\ V_j^i(\alpha) &:= \{v \in V;\ v_i = \alpha\}, && \text{für } i = (j-1)\cdot n + \hat{i},\\ & && \quad \hat{i} \in \{1,\dots,n\},\\ & && \quad j \in \{1,\dots,a\},\ \alpha \in \{\pm 1\}. \end{aligned}$$

- Zu einer beliebigen aber festen Wahl von $y \in \mathbb{R}^N$ definiere für $q \in \{1,\dots,Q\}$

$$\mu_q : S \to \mathbb{R},$$

$$s \mapsto \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{n}\left(y_{n(q-1)+j} - C_j(s)\right)^2\right) = \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(s)\right).$$

- Man berechne

$$\begin{aligned} &A_m(s), && \text{für } s \in S,\ m \in \{1,\dots,a-1\},\\ &B_m(s), && \text{für } s \in S,\ m \in \{1,\dots,Q\}, \end{aligned}$$

nach den oben angegeben Rekursionsformeln (57) und (70) und Startwerten $A_0(s)$, $B_0(s)$ mit (56) und (71).

- Man berechne alle A_α^i, $i \in \{1,\dots,K\}$, $\alpha \in \{\pm 1\}$ über

$$A_\alpha^i(y) = \sum_{s \in S} A_{j-1}(s) \sum_{t \in T(s, V_j^i(\alpha))} B_{Q-j+1}(t). \tag{83}$$

und bestimme die Soft-Outputs

$$L(U_i|y) = \ln\left(\frac{A_{+1}^i(y)}{A_{-1}^i(y)}\right), \quad i = 1,\dots,K.$$

Zusammen mit der Rekursionsformel aus dem vorangegangenen Abschnitt können alle $A_\alpha^i(y)$ jetzt gemeinsam mit $O(2^L \cdot Q)$ bzw. $O(K)$ Operationen statt $O(K2^K)$ Operationen berechnet werden.

Erinnerung: $L = k \cdot b$, $Q = a + k - 1$, $K = a \cdot b$, wobei a die Anzahl der Nachrichtenbits ist.

Die numerische Komplexität zur Berechnung der Soft-Outputs ist also von exponentieller Ordnung auf lineare Ordnung verringert worden, wobei a, die Anzahl der Nachrichtenbits, die entscheidende Größe ist.

Spezialfall: Binärer Zustandsübergang ($b = 1$)

Im wichtigen Spezialfall $b = 1$ besteht die Menge V der Zustandsübergangszeichen nur aus den beiden Elementen $+1, -1$. Die GSM-Codes gehören etwa zu diesem weit verbreiteten Spezialfall.

Da in der obigen Beschreibung jetzt $i = j$ und $V_j^i(\alpha) = \{\alpha\}$, vereinfacht sich die Vorgehensweise wie folgt:

- Definiere

$$V_j := \{\pm 1\}, \quad \text{für } j \in \{1, \ldots, a\},$$
$$V_j := \{+1\}, \quad \text{für } j \in \{a+1, \ldots, Q+1\}$$

- Zu einer beliebigen aber festen Wahl von $y \in \mathbb{R}^N$ definiere für $q \in \{1, \ldots, Q\}$

$$\mu_q : S \to \mathbb{R},$$
$$s \mapsto \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{n}\left(y_{n(q-1)+j} - C_j(s)\right)^2\right) = \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(s)\right).$$

- Man berechne

$$A_m(s), \quad \text{für } s \in S,\ m \in \{1, \ldots, a-1\},$$
$$B_m(s), \quad \text{für } s \in S,\ m \in \{1, \ldots, Q\},$$

nach den Rekursionsformeln (57) und (70) und Startwerten $A_0(s)$, $B_0(s)$ mit (56) und (71).

- Man berechne alle A_α^i, $i \in \{1, \ldots, K\}$, $\alpha \in \{\pm 1\}$ über

$$A_\alpha^i(y) = \sum_{s \in S} A_{i-1}(s) B_{Q-i+1}(T(s, \alpha)). \tag{84}$$

und bestimme die Soft-Outputs

$$L(U_i|y) = \ln\left(\frac{A_{+1}^i(y)}{A_{-1}^i(y)}\right), \quad i = 1, \ldots, K.$$

Algorithmische Umsetzung

Man betrachte für die algorithmische Umsetzung das Trellis-Diagramm

$$\mathcal{T} = \{(s,q);\ s \in S,\ q = 0,\ldots,Q+1\}$$

und die Abbildungen

- Knotengewichte im Zustand s des Trellis-Segments q

$$\begin{aligned}\mu:\ &\mathcal{T} \to \mathbb{R},\\ &(s,q) \mapsto \exp\left(-\frac{1}{2\sigma^2}\Delta F_q(s)\right)\end{aligned}$$

- Teilsummen 'A' im Zustand s des Trellis-Segments q

$$\begin{aligned}A:\ &\mathcal{T} \to \mathbb{R},\\ &(s,q) \mapsto A(s,q)\end{aligned}$$

- Teilsummen 'B' im Zustand s des Trellis-Segments $Q-q+1$

$$\begin{aligned}B:\ &\mathcal{T} \to \mathbb{R},\\ &(s,q) \mapsto B(s,q)\end{aligned}$$

Die Abbildungen werden nur auf den sinnvollen Teilmengen des Definitionsbereiches ausgewertet.

Fig.2 zeigt einen Algorithmus in Pseudocode-Notation, der ein Fortschreiten im Trellis-Diagramm unter Betrachtung aller Zustände zur Berechnung der Knotengewichte darstellt. Der Algorithmus veranschaulicht die obigen Ausführungen und ist aus sich heraus verständlich. Da der Wert von $\Delta F_q(s)$ nur mittelbar vom Zustand s abhängt und direkt mit $C(s)$ gebildet wird, gilt

$$|\{\Delta F_q(s);\ s \in S\}| \leq \min\left\{2^L, 2^n\right\},$$

d.h., für $n < L$ haben viele der obigen $\mu(s,q)$ den gleichen Wert. Abhängig vom speziellen Code läßt sich $\mu(s,q)$ in der Implementierung also mit weitaus weniger Operationen bestimmen.

Fig.3 und **Fig.4** zeigen je einen Algorithmus in Pseudocode-Notation zur Ermittlung von Softoutputs. Fig.3 bezieht sich auf den allgemeinen Fall und Fig.4 auf den Speziallfall für den binären Zustandsübergang ($b = 1$). Beide

Algorithmen veranschaulichen die obigen Ausführungen und sind aus sich heraus verständlich.

Bei geeigneter Implementierungsdarstellung von V bzw. $V_j^i(\alpha)$, etwa als Teilmengen von $\mathbb{N}$, lassen sich die obigen Iterationen $v \in V$ und $s \in S$ als gewöhnliche Programmschleifen implementieren. Vorkommende Indizes wie etwa $k - 1 + q$ werden bei der Implementierung natürlich nur einmal berechnet und nicht bei jedem Auftreten, wie es hier zur besseren Übersicht aufgeschrieben ist.

In **Fig.5** ist eine Prozessoreinheit PRZE dargestellt. Die Prozessoreinheit PRZE umfaßt einen Prozessor CPU, einen Speicher SPE und eine Input/Output-Schnittstelle IOS, die über ein Interface IFC auf unterschiedliche Art und Weise genutzt wird: Über eine Grafikschnittstelle wird eine Ausgabe auf einem Monitor MON sichtbar und/oder auf einem Drucker PRT ausgegeben. Eine Eingabe erfolgt über eine Maus MAS oder eine Tastatur TAST. Auch verfügt die Prozessoreinheit PRZE über einen Datenbus BUS, der die Verbindung von einem Speicher MEM, dem Prozessor CPU und der Input/Output-Schnittstelle IOS gewährleistet. Weiterhin sind an den Datenbus BUS zusätzliche Komponenten anschließbar, z.B. zusätzlicher Speicher, Datenspeicher (Festplatte) oder Scanner.

Patentansprüche

1. Verfahren zur Decodierung eines vorgegebenen Codeworts,

   (a) bei dem das Codewort mehrere Stellen mit unterschiedlichen Werten umfaßt;

   (b) bei dem jede Stelle des Codeworts einem Soft-Output-Wert zugeordnet wird, indem die Zuordnung basierend auf einer Trellis-Darstellung erfolgt;

   (c) bei dem durch die Zuordnung der einzelnen Stellen des Codeworts die Decodierung desselben bestimmt wird.

2. Verfahren nach Anspruch 1,
   bei dem die Berechnungsvorschrift für den Soft-Outpunt-Wert für jede Stelle des Codeworts bestimmt ist durch

$$L(U_i|y) = \ln\left(\frac{\sum\limits_{c\in\Gamma^i(+1)} \exp\left(-\frac{(y-c)^T(y-c)}{2\sigma^2}\right)}{\sum\limits_{c\in\Gamma^i(-1)} \exp\left(-\frac{(y-c)^T(y-c)}{2\sigma^2}\right)}\right), \quad \text{für } i = 1,\ldots,K\ ,$$

   wobei

$L(U_i\|y)$	ein Sicherheitsmaß (Soft-Output) für die i-te Stelle des zu bestimmenden Codeworts;
y	ein zu decodierendes Demodulationsergebnis;
c	ein Codewort;
$\Gamma^i(\pm 1)$	sämtliche Codewörter für $u_i = \pm 1$;
σ^2	eine Varianz (Kanalstörung)

   bezeichnen.

3. Verfahren nach Anspruch 2,
   bei dem die Decodiervorschrift gelöst wird, indem eine Eigenschaft eines Faltungscodes ausgenutzt wird, aus der hervorgeht, daß entsprechend einer Schieberegisteroperation Zustände bestimmt werden, aus denen wiederum die Trellis-Darstellung hervorgeht.

4. Verfahren nach einem der vorhergehenden Ansprüche,
   bei dem die Trellis-Darstellung ein vorgegebener Richtung bearbeitet wird, wobei ein Term $\tilde{A}_m$ durch Abbildungen A_m rekursiv ermittelt wird.

5. Verfahren nach Anspruch 4,
bei dem Abbildungen B_m anhand der Trellis-Darstellung bestimmt werden, wobei die Trellis-Darstellung entgegen der vorgegebenen Richtung durchlaufen wird.

6. Verfahren nach einem der Ansprüche 4 oder 5,
bei dem Terme A_α^i ermittelt werden, indem die Trellis-Darstellung erneut durchlaufen wird, wobei die bereits ermittelten Terme A_m und B_m berücksichtigt werden.

7. Verfahren nach einem der Ansprüche 4 bis 6, bei dem der Term $\tilde{A}_m$ bestimmt ist durch

$$\tilde{A}_m(E) = \sum_{s \in E} A_m(s), \quad \text{für } m \in \mathbb{N}$$

mit

$$A_m(s) = \mu_m(s) \sum_{t \in W(s, V_m)} A_{m-1}(t), \quad \text{für } m \in \mathbb{N}$$

und einem Startwert

$$A_0(s) = \begin{cases} 1 & : \text{ für } s = s_0, \\ 0 & : \text{ sonst.} \end{cases}$$

8. Verfahren nach einem der Ansprüche 5 bis 7,
bei dem der Term B_m bestimmt ist durch

$$B_m(s) = \mu_{Q-m+1}(s) \sum_{t \in T(s, V_{Q-m+2})} B_{m-1}(t), \qquad \text{für } 1 \leq m \leq Q,$$

wobei zur Terminierung der Rekursion

$$B_0(s) = \begin{cases} 1 & : \text{ für } s = s_0, \\ 0 & : \text{ sonst} \end{cases}$$

bestimmt wird.

9. Verfahren nach einem der Ansprüche 6 bis 8,
bei dem die Terme A_α^i ermittelt werden durch folgende Beziehung

$$A_\alpha^i(y) = \sum_{s \in S} A_{j-1}(s) \sum_{t \in T(s, V_j^i(\alpha))} B_{Q-j+1}(t).$$

10. Verfahren nach Anspruch 9,
bei dem die K Stellen des Codeworts bestimmt werden gemäß

$$L(U_i|y) = \ln\left(\frac{A^i_{+1}(y)}{A^i_{-1}(y)}\right), \quad i = 1, \ldots, K.$$

11. Verfahren nach einem der vorhergehenden Ansprüche,
zum Einsatz in einem Mobilfunknetz.

12. Verfahren nach Anspruch 11,
bei dem das Mobilfunknetz ein GSM-Netz ist.

13. Verfahren nach einem der vorhergehenden Ansprüche,
bei dem die Stellen des decodierten Codeworts in einer weiteren Decodierung einer binären Ergebnis, insbesondere einem Wert $+1$ oder einem Wert -1 zugeordnet werden, je nachdem welcher Wert näher am Ergebnis der ersten Decodierung liegt.

14. Anordnung zur Decodierung eines vorgegebenen Codeworts,
bei der eine Prozessoreinheit vorgesehen ist, die derart eingerichtet ist, daß

    (a) das Codewort mehrere Stellen mit unterschiedlichen Werten umfaßt;

    (b) jede Stelle des Codeworts einem Soft-Output-Wert zuordenbar ist, indem die Zuordnung basierend auf einer Trellis-Darstellung erfolgt;

    (c) durch die Zuordnung der einzelnen Stellen des Codeworts die Decodierung desselben bestimmbar ist.

Zusammenfassung

Verfahren und Anordnung zur Decodierung eines vorgegebenen Codeworts

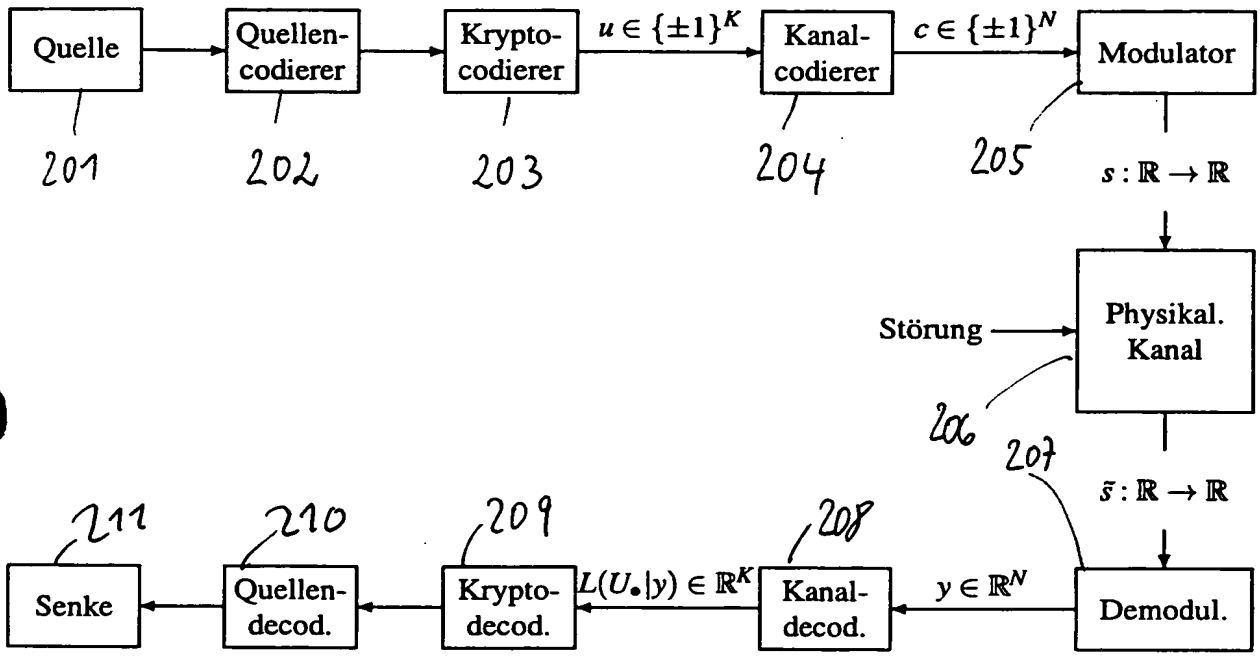
Es wird ein Verfahren zur Decodierung eines vorgegebenen Codeworts angegeben, bei dem das Codewort mehrere Stellen mit unterschiedlichen Werten umfaßt. Eine Encodierung erfolgt dabei insbesondere mit einem terminierten Faltungscode. Jeder Stelle des Codeworts wird ein Sicherheitsmaß (Soft-Output) für einen wahrscheinlichsten Booleschen Wert zugeordnet, indem die Zuordnung basierend auf einer Trellis-Darstellung durchgeführt wird. Durch die Zuordnung der einzelnen Stellen des Codeworts wird die Decodierung desselben ermittelt.

98P 5854

FIG 1

```
für q = 1,...,Q:
    für m = 1,...,n:
        j := n(q−1)+m;
        J_j := ∅;
        für r ∈ M_m:
            i := r + b(q−k);
            falls (i ≥ 1) ∧ (i ≤ K):
                J_j := J_j ∪ {i};
```

FIG 2



98P 5854

FIG 2

```
für q = 1,...,Q:
    für s ∈ S:
        μ(s,q) := exp(-1/(2σ²) ΔF_q(s));
```

FIG 3

```
für s ∈ S:                                                    Vorbelegung
    A(s,0) := 0; B(s,0) := 0;
A(s_0,0) := 1; B(s_0,0) := 1;                                 Startzustand
für q = 1,...,k:                                              Terminierung
    für s ∈ S:                                                Betrachtung aller Zustände
        s⁺ := T(s,v_0);                                       Nachfolgerzustand
        B(s,q) := μ(s,Q-(q-1))B(s⁺,q-1);                      Berechnung von B
für q = 1,...,a-1:                                            Fortschreiten im Trellis-Diagramm
    für s ∈ S:                                                Betrachtung aller Zustände
        A(s,q) := A(T̂(v_0,s),q-1);                            Vorbelegung von A
        B(s,k+q) := B(T(s,v_0),k-1+q);                        Vorbelegung von B
        für v ∈ V \ {v_0}:                                    Betrachtung aller Übergänge
            A(s,q) := A(s,q) + A(T̂(v,s),q-1);                 Berechnung von A
            B(s,k+q) := B(s,k+q) + B(T(s,v_0),k-1+q);         Berechnung von B
        A(s,q) := μ(s,q)·A(s,q);                              Berechnung von A
        B(s,k+q) := μ(s,a-q)·B(s,k+q);                        Berechnung von B
für i = 1,...,a:                                              Fortschreiten im Trellis-Diagramm
    A^i_{+1} := 0; A^i_{-1} := 0;                             Vorbelegung
    j = 1 + ⌊i/n⌋;
    für s ∈ S:                                                Betrachtung aller Zustände
        für v ∈ V^i_j(+1):                                    Übergänge
            A^i_{+1} := A^i_{+1} + A(s,j-1)·B(T(s,v),Q-j+1);   Update von A^i_{+1}
        für v ∈ V^i_j(-1):                                    Übergänge
            A^i_{-1} := A^i_{-1} + A(s,j-1)·B(T(s,v),Q-j+1);   Update von A^i_{+1}
    L(U_i|y) := ln(A^i_{+1}/A^i_{-1});                        i-ter Soft-Output
```

98P 5854

Fig. 4

```
für s ∈ S:                                                          Vorbelegung
    A(s,0) := 0; B(s,0) := 0;
└
A(s0,0) := 1; B(s0,0) := 1;                                         Startzustand
für q = 1,...,k:                                                    Terminierung
    für s ∈ S:                                                      Betrachtung aller Zustände
        s+ := T(s,+1);                                              Nachfolgerzustand
        B(s,q) := μ(s,Q−(q−1))B(s+,q−1);                            Berechnung von B
    └
└
für q = 1,...,a−1:                                                  Fortschreiten im Trellis-Diagramm
    für s ∈ S:                                                      Betrachtung aller Zustände
        t+ := T̂(+1,s); t− := T̂(−1,s);                              Vorgängerzustände
        s+ := T(s,+1); s− := T(s,−1);                               Nachfolgerzustände
        A(s,q) := μ(s,q)·(A(t+,q−1)+A(t−,q−1));                     Berechnung von A
        B(s,k+q) := μ(s,a−q)·(B(s+,k−1+q)+B(s−,k−1+q));             Berechnung von B
    └
└
für i = 1,...,a:                                                    Fortschreiten im Trellis-Diagramm
    A^i_{+1} := 0; A^i_{−1} := 0;                                   Vorbelegung
    für s ∈ S:                                                      Betrachtung aller Zustände
        s+ := T(s,+1); s− := T(s,−1);                               Nachfolgerzustände
        A^i_{+1} := A^i_{+1} + A(s,i−1)·B(s+,Q−i+1);                Update von A^i_{+1}
        A^i_{−1} := A^i_{−1} + A(s,i−1)·B(s−,Q−i+1);                Update von A^i_{−1}
    └
    L(Ui|y) := ln(A^i_{+1}/A^i_{−1});                               i-ter Soft-Output
└
```

98 P 5854

FIG 5